

QAQ: Bidirectional Semantic Coherence for Selecting High-Quality Synthetic Code Instructions

Jiayin Lei^{1*}, Ming Ma^{2,3*}, Yunxi Duan¹, Chenxi Li⁴, Tianming Yang^{2†}

¹School of Mathematics, Statistics and Mechanics, Beijing University of Technology, Beijing, 100124, China

²Institute of Neuroscience, State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, 200031, China

³School of Future Technology, University of Chinese Academy of Sciences, Beijing, 100049, China

⁴University of Chicago, Chicago, IL, USA

raylanxxsg559@gmail.com, {mam2022, tyang}@ion.ac.cn

Abstract

Synthetic data has become essential for training code generation models, yet it introduces significant noise and hallucinations that are difficult to detect with current metrics. Existing data selection methods like Instruction-Following Difficulty (IFD) typically assess how hard a model generates an answer given a query ($A|Q$). However, this metric is ambiguous on noisy synthetic data, where low probability can distinguish between intrinsic task complexity and model-generated hallucinations. Here, we propose QAQ, a novel data selection framework that evaluates data quality from the reverse direction: how well can the answer predict the query ($Q|A$)? We define Reverse Mutual Information (RMI) to quantify the information gain about the query conditioned on the answer. Our analyses reveal that both extremes of RMI signal quality issues: low RMI indicates semantic misalignment, while excessively high RMI may contain defect patterns that LLMs easily recognize. Furthermore, we introduce a selection strategy based on the disagreement between strong and weak models to identify samples that are valid yet challenging. Experiments across three datasets spanning code generation (WarriorCoder, Magpie-Qwen2.5-Coder-Pro-300K) and math reasoning (OpenR1-Math-220k) demonstrate that selecting just 25% of data using stratified RMI matches full-data performance while being consistently competitive with or better than existing data selection methods. Our approach highlights the importance of bidirectional semantic coherence in synthetic data curation, offering a scalable pathway to reduce computational costs without sacrificing model capability. Code is available at

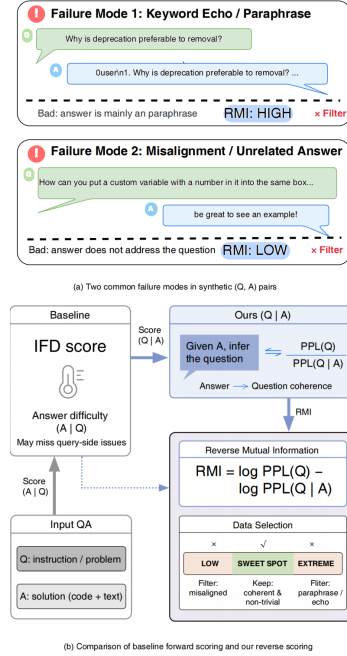


Figure 1: (a) Two common failure modes in synthetic (Q, A): keyword echo/paraphrase (high RMI) and misalignment (low RMI). (b) The comparisons between our method and traditional IFD method. Unlike previous work, QAQ scores question predictability given the answer ($Q|A$) and computes RMI. We keep the mid-range “sweet spot” and filter both extremes.

<https://github.com/XXSg559/QAQ>.

1 Introduction

Synthetic data has become the dominant paradigm for training code generation models (Chen et al., 2021; Roziere et al., 2023; Li et al., 2023). Seedless data synthesis methods like Magpie (Xu et al., 2025) and WarriorCoder (Feng et al., 2025) gener-

*Equal contribution. †Corresponding author.

ate diverse instruction-response pairs by prompting aligned LLMs directly, without relying on human-curated seed data. While this approach enables broad diversity, it also produces massive amounts of noisy data that are difficult to curate.

The quality issues in synthetic data are subtle and often undetectable by surface-level metrics. We observe two failure modes in practice: (1) queries containing fabricated terminology (e.g., “convert a gritty sulla matrix”), where models generate syntactically valid code that echoes these made-up terms, producing high surface-level quality scores while providing little learning signal; (2) off-topic queries that should not appear in a code dataset at all (e.g., a greeting “Hi!”), paired with incoherent responses (e.g., starting with “compensated...”)—indicating a completely broken Q-A pair. These failure modes are difficult for answer-focused metrics to detect.

Notably, methods like WarriorCoder focus on **answer quality**: multiple models generate competing solutions, and the highest-scored answer is selected for each query. However, this pipeline largely overlooks **query quality**.

Existing data selection methods like Instruction-Following Difficulty (IFD) (Li et al., 2024b) operate in the $A|Q$ direction. Although mutual information is theoretically symmetric, due to the autoregressive nature of language models, its practical estimation via conditional perplexity is not necessarily consistent. Empirically, we find low correlation ($\rho = 0.252$) between the two, confirming they capture distinct aspects of data quality.

We propose **QAQ** (Query, Answer-Query), a method that measures quality from the reverse direction: $\text{PPL}(Q|A)$, which captures how well the answer *explains* the question. The intuition is analogous to a student who struggles to understand a problem, but finds clarity after seeing the solution. We define RMI as $\log \text{PPL}(Q) - \log \text{PPL}(Q|A)$, which captures the information gain that the answer provides about the question:

- **Low RMI**: The answer provides little information gain about the question; seeing the answer does not help predict the question.
- **Very high RMI**: The answer makes the query trivially predictable, often due to superficial or shortcut patterns such as paraphrasing or repetition.

Furthermore, we observe that strong and weak models disagree on sample quality. Samples where

only the strong model recognizes high RMI may contain sophisticated patterns that weak models cannot capture. We call this the “cognitive gap”.

To test whether these ideas generalize beyond code, we additionally transfer QAQ to mathematical reasoning.

Our contributions are:

- We identify that seedless synthetic data pipelines overlook query quality, and propose QAQ, a data selection method based on Reverse Mutual Information (RMI) that captures query-answer semantic alignment.
- We empirically show that RMI serves as an effective quality indicator, where both misaligned and shortcut-dominated samples can be filtered through its value distribution.
- We introduce model disagreement as a selection signal, capturing samples with high “cognitive gap” between strong and weak models.
- We demonstrate that 25% of data selected by our method matches full-data performance while being consistently competitive with or better than existing data selection methods across both code generation and math reasoning domains.

2 Related Work

Data Selection for Instruction Tuning. Several approaches have been proposed to select high-quality samples from instruction-tuning datasets. IFD (Li et al., 2024b) measures sample quality using $\text{PPL}(A|Q)/\text{PPL}(A)$, the ratio of conditional to unconditional answer perplexity. Samples with IFD close to (but less than) 1 are preferred: $\text{IFD} \ll 1$ suggests the answer is trivially predictable from the instruction, while $\text{IFD} > 1$ indicates the instruction actually hinders answer generation. Superfiltering (Li et al., 2024a) shows that weak models can approximate strong models’ IFD scores, enabling efficient filtering at scale. These methods operate in the $A|Q$ direction, measuring generation difficulty rather than query-answer semantic alignment.

Beyond perplexity-based methods, gradient-based approaches like LESS (Xia et al., 2024) use influence estimation to identify training samples most relevant to target tasks. Multi-dimensional methods like D3 (Zhang et al., 2025) jointly optimize diversity, difficulty, and dependability via

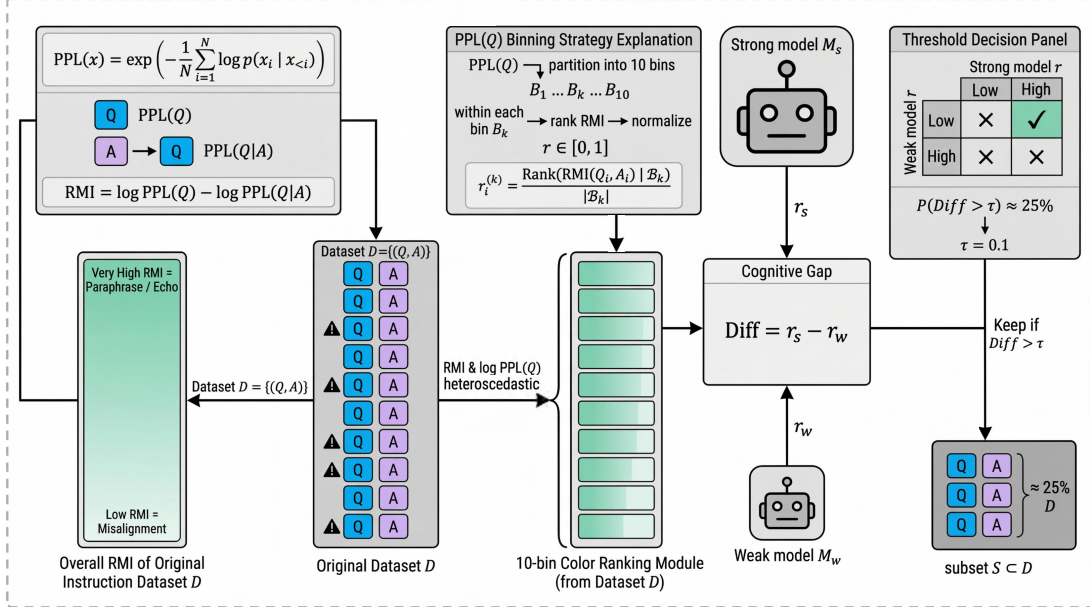


Figure 2: Overall workflow of the QAA framework for synthetic code data selection. **(Top-left)** RMI is calculated as the information gain provided by the answer about the query, using reverse perplexity. **(Bottom-left)** Our analysis identifies that RMI extremes signal quality issues: low RMI indicates semantic misalignment, while excessively high RMI indicates trivial paraphrasing or echoing. **(Center)** To eliminate the confounding factor of question complexity, RMI is stratified into 10 bins based on PPL(Q) to compute normalized within-bin ranks r . **(Right)** The final selection targets the “Cognitive Gap” (Diff = $r_s - r_w$) between a strong model M_s and a weak model M_w . Samples that are recognized as high-quality only by the strong model are retained, resulting in a high-quality 25% subset S that provides strong learning signals.

coreset objectives, while RDS+ (Iverson et al., 2025) selects samples by cosine similarity between pre-trained hidden-state representations and a small set of task-specific seed examples. Recent work explores orthogonal selection criteria: T-SHIRT (Fu et al., 2025) improves IFD through token-level selection and neighborhood robustness; SCAR (Li et al., 2025) targets style consistency between instructions and responses; ProDS (Guo et al., 2025) aligns selection with task-specific preferences via preference optimization. These methods optimize for diversity, style, or preference alignment, but do not directly assess query-answer semantic coherence.

Synthetic Data for Code Generation. High-quality datasets are essential for code generation. Code Alpaca (Chaudhary, 2023) and WizardCoder (Luo et al., 2024) use Evol-Instruct to evolve seed instructions into more complex programming problems. OSS-Instruct (Wei et al., 2024) generates diverse problems by extracting coding concepts from open-source code snippets. These seed-based methods inherit biases from their curated seeds, constraining diversity to the patterns present in the original data. In contrast, seedless approaches like

Magpie (Xu et al., 2025) and WarriorCoder (Feng et al., 2025) achieve greater diversity by prompting aligned LLMs directly without relying on human-curated data. However, this freedom comes at a cost: without seed constraints, the generated data contains more noise and semantic inconsistencies. While WarriorCoder selects the highest-scored answer for each query through cross-model evaluation, query quality itself remains largely unaddressed.

Mutual Information in NLP. Pointwise mutual information (PMI) has been widely used in NLP for measuring word associations and semantic similarity (Church and Hanks, 1990). At the token level, PMI helps identify collocations and construct semantic spaces. This concept can be extended to the sequence level, where mutual information between query and answer sequences reflects their semantic coherence. A related reverse-direction idea appears in summarization evaluation: Scialom et al. (2019) score a generated summary by how well it lets an automatic QA system answer questions derived from the source, using answerability as an unsupervised proxy for content overlap. We adopt a similar reverse-direction intuition, but apply it to

conditional perplexity for training data selection rather than post-hoc evaluation.

Model Comparison for Data Quality. Using multiple models to assess data has proven effective in various contexts. Helm et al. (2025) compare short-context and long-context models to weight tokens for long-range language modeling. Rho-1 (Lin et al., 2024) compares trained and untrained versions of the same model to identify tokens with high “excess loss” for selective pre-training. These works focus on token-level selection, while sample-level selection using model disagreement remains underexplored.

3 QAQ: Reverse Mutual Information for Data Selection

Given a large-scale instruction-tuning dataset $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^N$ for code generation, our goal is to select a high-quality subset $\mathcal{S} \subset \mathcal{D}$ that achieves comparable performance to training on the full dataset while significantly reducing computational costs.

3.1 Reverse Mutual Information

Prior work such as IFD (Li et al., 2024b) captures whether the instruction facilitates answer generation. Samples with IFD close to (but less than) 1 are preferred, representing what the model “does not yet know”.

We propose an alternative perspective: instead of measuring generation difficulty, we measure how well the answer *explains* the question. Specifically, we compute the **Reverse Mutual Information (RMI)** as:

$$\text{RMI}(Q, A) = \log \text{PPL}(Q) - \log \text{PPL}(Q|A) \quad (1)$$

where $\text{PPL}(Q)$ is the perplexity of the question alone, and $\text{PPL}(Q|A)$ is the perplexity of the question conditioned on the answer. Perplexity is computed as:

$$\text{PPL}(x) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log p(x_i | x_{<i}) \right) \quad (2)$$

where N is the number of tokens in sequence x .

This formulation is inspired by the mutual information $I(Q; A) = H(Q) - H(Q|A)$, capturing an *information-gain*-like quantity that reflects how much the answer helps predict the question under our scoring model. Intuitively:

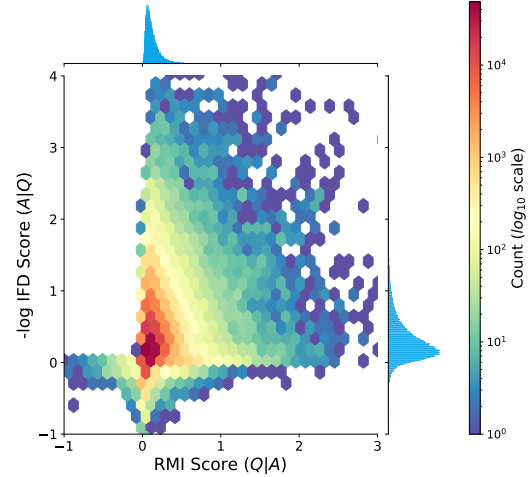


Figure 3: Scatter plot of RMI (Question-based) vs. $-\log(\text{IFD})$ (Answer-based). Despite both being mutual information estimates, the Spearman rank correlation is low ($\rho = 0.252$), suggesting they capture different aspects of data quality.

- High RMI indicates that seeing the answer makes the question much easier to predict, suggesting strong semantic coherence between Q and A .
- Low RMI suggests the answer does not relate well to the question (e.g., copy-paste errors, irrelevant responses).

Comparison with IFD. While IFD operates in the $A|Q$ direction, measuring “instruction-following difficulty”, our approach operates in the $Q|A$ direction, measuring “answer explanatory power”. We argue that a good code solution should clearly reflect the original problem specification: given the code, one should be able to infer what problem it solves.

Why the Two Directions Differ. As noted in the introduction, the $A|Q$ and $Q|A$ directions capture different signals due to length disparity, information density differences, and model capability asymmetry. This is consistent with the view of predictive $\mathcal{V}$ -information (Xu et al., 2020), which shows that for a bounded predictor, conditioning in opposite directions generally yields different amounts of usable information, unlike symmetric Shannon mutual information. Figure 3 shows the low correlation ($\rho = 0.252$) between RMI and IFD, confirming they measure distinct aspects of data quality.

Implementation of $\text{PPL}(Q)$. We compute $\text{PPL}(Q)$ using the standard chat template, which

matches the format used during training:

System: Standard system prompt
User: [Q]

We compute $\text{PPL}(Q)$ on the question tokens in the user turn via teacher-forcing with per-token length normalization.

Implementation of $\text{PPL}(Q|A)$. A naive implementation would simply concatenate the answer before the question, but this creates an unnatural input order. Instead, we frame the computation as a *reverse generation task* using the chat template:

System: Standard system prompt
User: TASK: Given an answer, generate the most likely computer science question that this answer is responding to. If the inferred question is outside computer science, respond with “INVALID”. Answer: [A]
Assistant: [Q]

We compute $\text{PPL}(Q|A)$ via teacher-forcing on the ground-truth question tokens in the assistant turn, with per-token length normalization. While the two computations use different message roles (user vs. assistant), the RMI score captures how much the answer A reduces uncertainty about Q . The relative difference is meaningful even across different positional contexts. Full prompt templates and system configurations are provided in Appendix D.

Validation of Chat Template. We verify that IFD remains highly robust across template variations ($\rho = 0.957$). In contrast, RMI is sensitive to context: removing the template causes a “cold start” in $\text{PPL}(Q)$ calculation, where the lack of context artificially inflates perplexity and biases the score. This suggests that template context is important for RMI to capture genuine semantic coherence. We provide a detailed sensitivity analysis in Appendix E.

3.2 Stratified RMI

Directly filtering by global RMI rankings introduces a confounding factor: question complexity. Simple questions naturally have low $\text{PPL}(Q)$, leading to lower RMI scores regardless of answer quality. As shown in Figure 4, the relationship between RMI and $\log \text{PPL}(Q)$ exhibits clear heteroscedasticity: samples with low $\text{PPL}(Q)$ have narrowly distributed RMI values (concentrated at lower ranges), while samples with high $\text{PPL}(Q)$

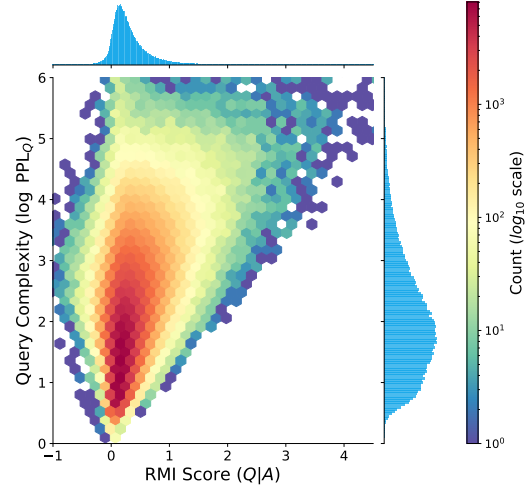


Figure 4: Scatter plot of RMI vs. $\log \text{PPL}(Q)$. The heteroscedastic pattern (low variance at low PPL, high variance at high PPL) motivates stratification by question complexity.

show much greater RMI variance. This indicates that RMI and question complexity capture different aspects of data quality. Without stratification, simple but valid questions may be incorrectly penalized due to their inherently lower RMI ceiling.

To address this, we propose **Stratified RMI**: we first partition the dataset into $K = 10$ bins (deciles) based on $\text{PPL}(Q)$, then compute RMI rankings independently within each bin. We use $K = 10$ as a default, which our ablation study (Section 4.3) shows to be a stable choice across a range of values. This ensures that samples are compared against others of similar question complexity.

Formally, let $\mathcal{B}_k$ denote the k -th bin containing samples with similar $\text{PPL}(Q)$ values. For each sample $(Q_i, A_i) \in \mathcal{B}_k$, we compute its within-bin rank and normalize to $[0, 1]$:

$$r_i^{(k)} = \frac{\text{Rank}(\text{RMI}(Q_i, A_i) \mid \mathcal{B}_k)}{|\mathcal{B}_k|} \quad (3)$$

where $|\mathcal{B}_k|$ is the number of samples in bin k .

3.3 Model Disagreement

As we show empirically in Section 4.3, mid-range RMI outperforms both extremes, yet manually specifying such percentile boundaries is arbitrary and dataset-dependent. We therefore seek a more principled selection criterion. Following prior work showing that weak models can approximate strong models’ data quality scores (Li et al., 2024a), we first evaluate whether RMI exhibits similar model-invariance. While strong and weak models show

high rank correlation (Spearman $\rho \approx 0.7$) and 76.1% overlap in top-50% selection, their downstream performance differs substantially (validated in Section 4.3). This suggests the non-overlapping samples carry disproportionate training signal.

We therefore explore **model disagreement** as an alternative signal. We compute stratified RMI ranks using two models:

- A strong model M_s (e.g., DeepSeek-Coder-6.7B-Base)
- A weak model M_w (e.g., Qwen3-0.6B)

For each sample, we obtain two normalized ranks $r_s, r_w \in [0, 1]$ and define:

$$\text{Diff} = r_s - r_w \quad (\text{disagreement signal}) \quad (4)$$

High Diff indicates samples where the strong model gives high RMI rank while the weak model gives low RMI rank. We hypothesize that such samples are both *valid* (recognized by strong model) and *learnable* (challenging for weak model). We validate this hypothesis empirically in Section 4.3. Our final selection retains the top 25% of samples ranked by Diff score. We observe that this cutoff falls at approximately $\text{Diff} > 0.1$ on WarriorCoder.

4 Experiments

4.1 Experimental Setup

Dataset. We use a reproduction of the WarriorCoder dataset (Feng et al., 2025) from HuggingFace, containing 329K instruction-response pairs for code generation created through multi-agent collaboration. After filtering samples exceeding 2048 tokens, approximately 310K samples remain. The original dataset is not publicly available.

Models. For computing RMI scores, we use:

- Strong model: DeepSeek-Coder-6.7B-Base (Guo et al., 2024)
- Weak model: Qwen3-0.6B (Team, 2025)

For fine-tuning evaluation, we train DeepSeek-Coder-6.7B-Base on selected subsets.

Training Details. We fine-tune for 3 epochs using LlamaFactory (Zheng et al., 2024). Due to the noisy nature of our dataset, we refrain from excessively reducing the batch size. Batch sizes and learning rates are adjusted by dataset size: for full

data, we use batch size 512 and learning rate $1.2e-4$; for 50% data, batch size 256 and learning rate $0.8e-4$; for 25% data, batch size 256 and learning rate $0.4e-4$. Additionally, the learning rate scheduler follows a cosine decay with a warmup ratio of 0.2.

Evaluation. We evaluate on HumanEval (Chen et al., 2021), HumanEval+, MBPP (Austin et al., 2021), and MBPP+ benchmarks (Liu et al., 2023) using greedy decoding.

Comparisons. We include two groups. **RMI ablation variants** vary the selection range and model choice:

- RMI Top 50%: Top 50% by strong model (DeepSeek-Coder-6.7B-Base) stratified RMI
- RMI QW 50%: Top 50% by weak model (Qwen3-0.6B) stratified RMI
- RMI 50-75%: Middle range by strong model stratified RMI
- RMI Top/Bottom-25%: Extreme ranges by strong model stratified RMI

External baselines represent competing data selection methods, spanning perplexity-based, representation-based, and style-based selection mechanisms:

- Random: Uniformly sample 25% of the data without replacement.
- IFD (Li et al., 2024b): Computes $\text{PPL}(A|Q)/\text{PPL}(A)$ using DeepSeek-Coder-6.7B-Base and retains the top 25% of samples with IFD score closest to (but below) 1.
- RDS+ (Iverson et al., 2025): Each (Q, A) pair is embedded using the base model’s own hidden states via a position-weighted mean pool of the last hidden layer (later tokens weighted more heavily), and cosine similarity is computed against a small set of task-specific seed examples (25 from HumanEval, 25 from MBPP). Samples are selected round-robin across the seed examples, taking the nearest unselected neighbor for each seed in turn, until the 25% budget is reached.
- SCAR (Li et al., 2025): A trained style-consistency ranker

Method	%	HumanEval	HumanEval+	MBPP	MBPP+
Full Data	100	78.05	72.56	71.69	59.52
RMI Top-50%	50	78.05	73.17	72.22	58.20
RMI 50% (G)	50	76.22	70.12	73.02	60.58
RMI QW 50%	50	72.56	68.90	70.63	58.47
RMI 50-75%	25	77.44	72.56	71.43	58.47
RMI Top-25%	25	73.17	70.12	70.11	57.14
RMI Bot-25%	25	72.56	67.68	65.34	54.76
Random	25	73.78	69.51	68.52	57.67
IFD	25	71.95	66.46	64.81	54.76
RDS+	25	76.83	71.34	71.69	58.99
SCAR	25	75.00	70.73	70.63	57.67
QAQ (Ours)	25	77.44	71.95	71.43	58.73

Table 1: Main results (pass@1 with greedy decoding). (G) = global ranking. **QAQ (Ours)** selects the top 25% of samples by Diff score ($\text{Diff} = r_s - r_w$); see Section 4.3.

(lizhuang144/scar-gte-large) scores each (Q, A) pair on two axes: linguistic form (surface structure of the response, e.g. sentence structure and punctuation, independent of the instruction) and instructional surprisal (how closely $\text{PPL}(A|Q)$ matches human-written response patterns). The top 25% by combined score are retained.

All perplexity computations use the same chat template format for fair comparison.

Math Domain. To verify generalization beyond code, we apply QAQ to OpenR1-Math-220k (Hugging Face, 2025) (91K samples after filtering to 16,384-token cutoff), using Qwen2.5-Math-7B-Instruct (Yang et al., 2024) as the base model and following OpenR1-Qwen-7B’s training configuration (Hugging Face, 2025). All methods select 25% of data; QAQ uses Qwen2.5-Math-7B and Qwen2.5-Math-1.5B (Yang et al., 2024) as strong and weak models with the same 10-bin stratification. We evaluate on MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021) and GPQA-Diamond (Rein et al., 2024) (Pass@1) using the LightEval framework (Habib et al., 2023).

4.2 Main Results

We fine-tune DeepSeek-Coder-6.7B-Base on selected subsets from WarriorCoder (310K samples) and evaluate on HumanEval(+) and MBPP(+) (Liu et al., 2023). Results are shown in Table 1.

Key findings:

- **RMI outperforms most baselines:** At 25% data, RMI 50-75% outperforms Random, IFD, and SCAR on both benchmarks. It achieves

results comparable to RDS+, surpassing it on HumanEval+ (72.56 vs 71.34) while remaining competitive on MBPP+.

- **Middle range is optimal:** RMI 50-75% (25% data) matches full-data performance on HumanEval+ (72.56) and approaches it on MBPP+ (58.47 vs 59.52). This “sweet spot” contains samples with good semantic alignment without trivial paraphrasing.
- **Both extremes are harmful:** RMI Top-25% underperforms full data by 2.4/2.4 points on HumanEval+/MBPP+ (70.12/57.14 vs 72.56/59.52), suggesting extremely high RMI indicates certain defect patterns that language models easily recognize, such as answer paraphrasing the query. RMI Bottom-25% underperforms by 4.9/4.8 points (67.68/54.76), confirming low RMI indicates semantic misalignment.
- **Model disagreement further improves HumanEval:** QAQ (25% data, selected by model disagreement) achieves the highest HumanEval (77.44) among all external baselines at 25%, outperforming RDS+ on both HumanEval and HumanEval+ while maintaining competitive MBPP performance.

Case Study: Defect Patterns in High RMI. To illustrate why extremely high RMI samples are harmful, we examine an example from the top RMI percentile ($\text{RMI} = 3.37$):

Query: How to convert a gritty sulla matrix with floating discharges into a fairly crisp supra array predominantly within specific thresholds using Python?
Answer: Converting a Gritty Sulla Matrix... To convert a gritty sulla matrix with floating discharges into a fairly crisp supra array... (followed by generic NumPy thresholding code)

The query contains nonsensical jargon, and the answer simply paraphrases these terms (underlined). The answer makes the query highly predictable (hence high RMI) but provides little genuine learning signal.

Case Study: Low RMI as Misalignment. Conversely, we examine a sample with negative RMI ($\text{RMI} = -0.13$):

Method	Data Size	MATH-500	GPQA-D
Full Data [†]	100%	90.6	42.4
Random	25%	87.8	33.3
IFD	25%	87.2	37.4
SCAR	25%	86.6	30.8
RDS+	25%	85.6	37.9
QAQ (Ours)	25%	91.6	39.9

Table 2: Results on math domain benchmarks. [†]Full Data is the official OpenR1-Qwen-7B checkpoint (Hugging Face, 2025) (Qwen2.5-Math-7B-Instruct base).

Query: Hi!

Answer: compensated, let’s get you the help you need. Could you please provide more details about what you’re working on or what specific Python coding question you have? I’m here to assist you with any Python-related challenges you might be facing. Whether it’s a beginner question, a complex algorithm, or anything in between, feel free to share the details, and I’ll do my best to help!

The query “Hi!” is not a meaningful instruction for a code dataset. Consequently, the generated answer—despite attempting to be helpful—cannot provide information that helps predict the query. This is a typical example of **semantic misalignment**: a poorly-specified query leads to an answer that has little semantic connection to the query content (hence negative RMI). Such samples should be filtered out as they provide poor learning signal for code generation.

Math Domain Results. Table 2 reports results on MATH-500 and GPQA-Diamond. Using only 25% of the data, QAQ approaches full-data performance on both benchmarks (91.6 vs. 90.6 on MATH-500; 39.9 vs. 42.4 on GPQA-D) while outperforming every other 25%-data baseline, demonstrating that RMI generalizes beyond code generation. All baselines fall further below the full-data performance than QAQ on both benchmarks, confirming that naive data reduction without quality-aware selection is more harmful in the math domain.

4.3 Analysis

Why Does Model Disagreement Work? A single model’s RMI conflates multiple factors that cannot be easily separated:

- **Low RMI** may indicate: (1) the sample is too *difficult* for the model to understand the Q-A relationship; (2) the Q-A pair is genuinely *unrelated*; (3) the data is *corrupted*; or (4) the model is already *familiar* with this Q-A pat-

Strong	Weak	Action	Interpretation
High	High	Filter	Defect patterns (e.g., repetition)
Low	Low	Filter	Corrupted, too hard, or trivial
Low	High	Filter	Easy for strong model
High	Low	Retain	Valid (strong) + learnable (weak)

Table 3: Model agreement/disagreement cases.

Method	HumanEval	HumanEval+	MBPP	MBPP+
Diff-High	77.44	71.95	71.43	58.73
Sum-High	74.39	68.90	71.16	59.52
Sum-Low	71.34	65.85	66.14	55.56
Diff-Low	71.34	67.07	74.87	62.43

Table 4: Ablation on selection strategies (all 25% data). Diff-High corresponds to QAQ (Ours) in Table 1.

tern, so the answer provides little additional information.

- **High RMI** may indicate: (1) the model recognizes a familiar pattern from pretraining; or (2) defect patterns such as paraphrasing or repetition.

By comparing models of different capabilities, we disentangle these signals: the strong model ensures *validity* while the weak model ensures *learnability*. Table 3 summarizes the four cases.

This explains why Diff-based selection outperforms both single-model selection and consensus-based selection (Sum-High).

Ablation: Disagreement vs. Consensus. We compare four selection strategies under fixed hyperparameters (batch size 256, lr 4e-5, warmup 0.2); Table 4 reports best-checkpoint results.

Key observations: (1) **Diff > Sum**: Diff-High outperforms Sum-High by 3 points on HumanEval+ (71.95 vs. 68.90), confirming disagreement is more informative than consensus. (2) **Sum-Low is worst**: Both-low agreement yields the poorest results, confirming such samples should be filtered. (3) **Diff-High vs Diff-Low**: Diff-High favors HumanEval while Diff-Low favors MBPP, suggesting the two signals capture complementary difficulty levels.

Stability of RMI Across Training. Pre- and post-finetuning RMI rankings correlate at $\rho = 0.9539$, confirming that RMI captures intrinsic data properties and is suitable for static one-time selection. The moderate cross-model correlation

($\rho = 0.70$) shows that RMI is also sensitive to model capability, which is precisely what enables the Diff signal. We discuss model-specific bias in Appendix F.

Sample Overlap Analysis. Diff-High and Sum-High select fundamentally different samples: their overlap is only **13.85%**, explaining the 3-point HumanEval+ gap (71.95 vs. 68.90). Sum-High likely captures many paraphrasing cases—repetition is a simple pattern that both models detect, yielding high r_s and r_w simultaneously (Holtzman et al., 2020)—whereas Diff-High retains samples with genuine semantic content that only the strong model recognizes. Separately, strong and weak model top-50% selections overlap at 76.1%, yet yield different downstream performance, confirming that the 24% non-overlapping samples carry disproportionate training signal.

We further verify generalization to Magpie-Qwen2.5-Coder-Pro-300K (Appendix A), robustness to model pair choice (Appendix B), and stability across bin counts $K \in \{5, 10, 20, 50\}$ (Appendix C).

5 Conclusion

We proposed QAQ, a data selection framework that evaluates quality from the reverse direction $\text{PPL}(Q|A)$ and leverages model disagreement to identify samples that are both valid and learnable, demonstrating its effectiveness across code generation and math reasoning.

Our experiments reveal four key findings: (1) RMI-based selection with stratification consistently outperforms IFD-based selection, validating the importance of reverse direction analysis; (2) RMI outliers at both ends correlate with undesirable samples; (3) Model disagreement captures a unique subset of valuable samples (only 13.85% overlap with consensus-based selection), yielding a 3-point HumanEval+ gain over consensus-based selection; (4) The method generalizes across domains: on math reasoning (OpenR1-Math-220k), QAQ approaches full-data performance with 25% of the data while outperforming other 25%-data baselines on both MATH-500 (91.6 vs. 90.6) and GPQA-Diamond (39.9 vs. 42.4), demonstrating that RMI is a quality signal that transfers across the two domains we tested (code generation and math reasoning).

Practically, selecting 25% of data with QAQ achieves competitive performance on code gener-

ation and approaches full-data performance while outperforming other selection methods on math reasoning, offering a scalable path to reduce fine-tuning cost without sacrificing capability.

Limitations

Dataset Diversity. Main experiments use WarriorCoder, with validation on Magpie-Qwen2.5-Coder-Pro-300K (Appendix A) and OpenR1-Math-220k (Section 4.1). Both code datasets are seedless synthetic and may share noise patterns; future work should cover seed-based corpora and general instruction-tuning datasets.

Model Selection. The disagreement signal depends on the model pair. We use DeepSeek-Coder-6.7B-Base and Qwen3-0.6B; Appendix B shows robustness across three pairings including a non-code model, though the optimal pair for other domains or languages remains unexplored.

Hyperparameters. We use $K = 10$ as a default; ablations (Appendix C) confirm stability across $K \in \{5, 10, 20, 50\}$, suggesting the method is robust to this choice.

Reproducibility. Results are single-run; future work should report multi-seed confidence intervals. Checkpoint selection is also not uniform across domains: training on the noisy WarriorCoder data is less stable across steps, so code-domain results select the best checkpoint, whereas math-domain training (following the OpenR1-Qwen-7B recipe) is more stable across steps, so we report the final training step for all methods.

Computational Cost. RMI requires two forward passes per sample—a one-time overhead. See Appendix G for timing details.

Acknowledgments

This work was supported by the Brain Science and Brain-like Intelligence Technology-National Science and Technology Major Project (grant no. 2021ZD0203702) to T.Y.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. <https://github.com/sahil280114/codealpaca>.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Huawen Feng, Pu Zhao, Qingfeng Sun, Can Xu, Fangkai Yang, Lu Wang, Qianli Ma, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2025. **WarriorCoder: Learning from expert battles to augment code large language models**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4955–4969, Vienna, Austria. Association for Computational Linguistics.

Yanjun Fu, Faisal Hamman, and Sanghamitra Dutta. 2025. **T-shirt: Token-selective hierarchical data selection for instruction tuning**. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 113932–113958. Curran Associates, Inc.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y.K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. **Deepseek-coder: When the large language model meets programming – the rise of code intelligence**. *Preprint*, arXiv:2401.14196.

Wenya Guo, Zhengkun Zhang, Xumeng Liu, Ying Zhang, Ziyu Lu, Haoze Zhu, Xubo Liu, and Ruxue Yan. 2025. **Prods: Preference-oriented data selection for instruction tuning**. *arXiv preprint arXiv:2505.12754*.

Nathan Habib, Cl  mentine Fourier, Hynek Kydl  cek, Thomas Wolf, and Lewis Tunstall. 2023. **Lighteval: A lightweight framework for llm evaluation**.

Falko Helm, Nico Daheim, and Iryna Gurevych. 2025. **Token weighting for long-range language modeling**. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1440–1459, Albuquerque, New Mexico. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. **Measuring mathematical problem solving with the math dataset**. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. **The curious case of neural text degeneration**. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hugging Face. 2025. **Open-r1: A fully open reproduction of deepseek-r1**. <https://github.com/huggingface/open-r1>.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025. **Large-scale data selection for instruction tuning**. *arXiv preprint arXiv:2503.01807*.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. **Superfiltering: Weak-to-strong data filtering for fast instruction-tuning**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. **From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Raymond Li, Loubna Ben allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia LI, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Joel Lamy-Poirier, Joao Monteiro, Nicolas Gontier, Ming-Ho Yee, and 39 others. 2023. **StarCoder: may the source be with you! Transactions on Machine Learning Research. Reproducibility Certification**.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. 2025. **SCAR: Data selection via style consistency-aware response ranking for efficient instruction-tuning of large language models**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12790, Vienna, Austria. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. **Let's verify step by step**. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. **Not all tokens are what you need for pretraining**. In *Advances in Neural Information Processing Systems*, volume 37, pages 29029–29063. Curran Associates, Inc.

- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. [Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 21558–21572. Curran Associates, Inc.
- Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2024. [Wizardcoder: Empowering code large language models with evol-instruct](#). In *International Conference on Learning Representations*, volume 2024, pages 27168–27188.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.
- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. [Answers unite! unsupervised metrics for reinforced summarization models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.
- Qwen Team. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. [Magicoder: Empowering code generation with OSS-instruct](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52632–52657. PMLR.
- Mengzhou Xia, Sathika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. [LESS: Selecting influential data for targeted instruction tuning](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54104–54132. PMLR.
- Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. [A theory of usable information under computational constraints](#). In *International Conference on Learning Representations*.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. [Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing](#). In *International Conference on Learning Representations*, volume 2025, pages 76346–76382.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.
- Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. 2025. [D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient llm instruction tuning](#). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 8348–8356. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 2024. [LlamaFactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Generalization to Magpie-Qwen2.5-Coder-Pro-300K

To verify generalization beyond WarriorCoder, we evaluate on Magpie-Qwen2.5-Coder-Pro-300K (Xu et al., 2025), another seedless synthetic dataset. After filtering samples exceeding 2048 tokens, approximately 120K samples remain. We use the same model pair and select 25% of data.

Method	%	HumanEval	HumanEval+	MBPP	MBPP+
Full Data	100	71.95	66.46	62.43	53.17
Random	25	72.56	65.24	63.76	53.17
IFD	25	67.07	60.98	61.64	49.21
SCAR	25	73.78	66.46	63.23	49.74
RDS+	25	70.73	64.02	69.58	57.67
QAQ (Ours)	25	71.95	65.24	68.25	56.35

Table 5: Generalization to Magpie-Qwen2.5-Coder-Pro-300K. QAQ achieves the most balanced results across all four benchmarks; SCAR underperforms Full Data on MBPP+ (49.74 vs. 53.17), indicating benchmark-specific overfitting.

QAQ demonstrates competitive and balanced performance across all four benchmarks, without sacrificing any single metric. SCAR leads on HumanEval but drops below Full Data on MBPP+, suggesting benchmark-specific overfitting. RDS+ leads on MBPP/MBPP+ but underperforms on HumanEval+.

B Sensitivity to Model Pair

To examine whether the cognitive gap signal depends on the specific model pair, we test two configurations beyond our original pairing: (1) DeepSeek-Coder-6.7B vs. Qwen2.5-Coder-7B (a stronger code model), and (2) DeepSeek-Coder-6.7B vs. Mistral-7B-v0.3 (a non-code-specialized model with a different architecture). All use Diff-High selection at 25% data.

Weak Model	HumanEval	HumanEval+	MBPP	MBPP+
Full Data (100%)	78.05	72.56	71.69	59.52
Qwen3-0.6B (orig.)	77.44	71.95	71.43	58.73
Qwen2.5-Coder-7B	77.44	72.56	73.02	60.32
Mistral-7B-v0.3	75.00	69.51	74.60	60.85

Table 6: Sensitivity to model pair (strong model fixed as DeepSeek-Coder-6.7B-Base; 25% data, Diff-High).

Two findings stand out. First, Qwen2.5-Coder-7B is a stronger code model than DeepSeek-Coder-6.7B, yet pairing them matches or exceeds Full Data on three of four metrics (HumanEval+, MBPP,

MBPP+)—showing that the method works even when the designated “weak” model is actually stronger. Second, Mistral-7B-v0.3, a non-code-specialized model with a different architecture, still yields effective selection. Together, these results indicate that the key factor is *meaningful disagreement* between the two models, rather than a strict capacity gap.

C Sensitivity to Binning Parameter K

The number of bins K primarily affects mid-range selection: a coarser partition mixes samples across a wider complexity range within each bin, while a finer partition enforces stricter complexity matching but reduces within-bin sample size. Table 7 shows that performance is stable across $K \in \{5, 10, 20, 50\}$, with $K = 5$ –10 performing best overall, confirming the method is robust to this hyperparameter.

K	HumanEval	HumanEval+	MBPP	MBPP+
5	78.66	70.73	74.07	60.58
10	77.44	72.56	71.43	58.47
20	74.39	70.73	73.28	60.58
50	75.00	70.73	73.28	60.05

Table 7: Ablation on number of PPL(Q) bins K (RMI 50-75%, 25% data).

D Implementation Details and Model Configurations

To provide full transparency on how Reverse Mutual Information (RMI) is calculated, we present the exact chat templates and system prompts used for both the strong and weak models.

DeepSeek-Coder Template The following template follows the official convention for DeepSeek-Coder-Instruct.

System Template for DeepSeek-Coder

System: You are an AI programming assistant, utilizing the Deepseek Coder model, developed by Deepseek Company, and you only answer questions related to computer science. For politically sensitive questions, security and privacy issues, and other non-computer science questions, you will refuse to answer.

Qwen3 Template For Qwen3-0.6B, we use a consistent system prompt to focus the model on computer science tasks. Note that for this model, we explicitly set `enable_thinking=False` to maintain standard autoregressive behavior.

System Template for Qwen3

System: You are an AI programming assistant, and you only answer questions related to computer science. For politically sensitive questions, security and privacy issues, and other non-computer science questions, you will refuse to answer.

Implementation Note In both cases, $PPL(Q|A)$ is calculated by performing a forward pass on the ground-truth question tokens represented by $[Q]$ in the Assistant turn, while teacher-forcing the System and User inputs as context.

E Sensitivity to Chat Template and Task Framing

We investigate how specific task instructions and formatting affect RMI and IFD signals.

Standard Implementations (Template-based).

In our main experiments, both metrics utilize structured chat templates. For ****Standard RMI****, the reverse generation task is highly constrained: “TASK: Given an answer, generate the most likely **computer science** question that this answer is responding to. If the inferred question is outside computer science, respond with ‘INVALID’.”

For ****Template-based IFD****, we compute the perplexity of the answer A within the following structured templates:

System: Standard system prompt
User: [None]
Assistant: [A]

System: Standard system prompt
User: [Q]
Assistant: [A]

Variations for Sensitivity Analysis. To isolate the impact of task framing, we compare our standard method against several variations:

- **Raw IFD:** Following the original implementation proposed by Li et al. (2024b), Raw IFD is computed via simple text concatenation: $[Q] + [A]$ for the conditional part, and $[A]$ for the unconditional part, without any system prompts or role indicators (User/Assistant).
- **Simplified Task Prompt** ($\rho = 0.88$): Removes both the domain specificity and the negative constraint, using a more general instruction: “TASK: Given an answer, generate

the most likely question that this answer is responding to.”

- **System Context Only** ($\rho = 0.79$): For $PPL(Q)$, we retain the System Prompt but **remove the explicit “TASK” instruction**. This tests whether the model’s persona alone (e.g., “You are an AI assistant”) is sufficient for estimating query probability.
- **Prefix-only** ($\rho = 0.70$): For $PPL(Q)$, we strip away the System Prompt and use only a minimal newline-delimited indicator (“Question:”).
- **Pure Raw** ($\rho = 0.62$): For $PPL(Q)$, we calculate the perplexity from the very first token of the query with **zero context** (cold start).

Results and Analysis. As shown in Table 8, IFD is remarkably stable ($\rho = 0.957$) even under raw concatenation. In contrast, RMI drops as we strip away the template and context. The correlation decreases from 0.88 (Simplified Template) to 0.79 (System-Prefix), 0.70 (Prefix-only), and 0.62 (Pure Raw). This confirms that while the specific *wording* of the template matters less, the presence of **sufficient context** for $PPL(Q)$ is critical. Even without a task template, providing the system prompt ($\rho = 0.79$) significantly stabilizes the metric compared to raw inputs.

Comparison (vs. Standard RMI)	Spearman ρ
IFD vs. Raw IFD	0.957
RMI vs. Simplified Task Prompt	0.88
RMI vs. System-Prefix (No Template)	0.79
RMI vs. Prefix-only (No Template)	0.70
RMI vs. Pure Raw (No Template)	0.62

Table 8: Spearman correlation of metrics under different task framings.

The primary source of instability is the **contextual imbalance in $PPL(Q)$ calculation**. In the *Pure Raw* setting, the model must predict the query from a “cold start” without knowing it is handling a CS query. Since queries are typically short, the high loss incurred by the initial tokens dominates the average $PPL(Q)$, biasing the *RMI* upwards and introducing noise (see Figure 5).

Discussion: Why IFD is More Robust. IFD remains robust because $PPL(A)$ is calculated on answer sequences (code) that are typically longer and

follow familiar structural patterns (def, import). Moreover, A is always conditioned on Q in the IFD calculation, avoiding the "zero context" problem. The correlation jump from 0.62 to 0.70 by simply adding a "Question:" indicator proves that task framing is essential for RMI to capture genuine semantic coherence rather than prefix-induced surprise.

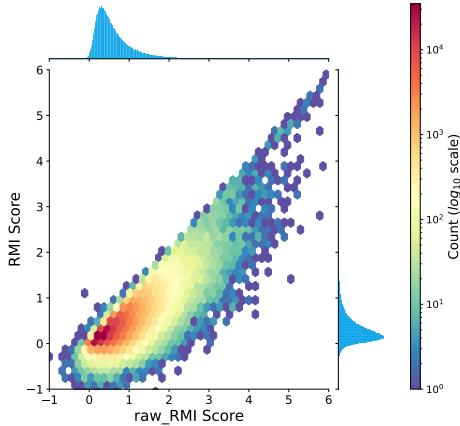


Figure 5: Scatter plot comparing Standard RMI vs. Raw RMI. The lack of system context in the Raw version leads to inflated $PPL(Q)$ and subsequent overestimation of RMI.

Table 9 further confirms this: Standard RMI 50-75% outperforms Simplified RMI by 2–3 points on HumanEval(+), and both RMI variants substantially outperform IFD even when IFD uses an enhanced semantic-constrained prompt. This confirms that the core effectiveness of QAQ comes from the reverse direction ($A \rightarrow Q$) rather than prompt engineering.

Method	HumanEval	HumanEval+	MBPP	MBPP+
Standard RMI 50-75%	77.44	72.56	71.43	58.47
Simplified RMI 50-75%	75.00	69.51	70.37	58.20
IFD w/ enhanced prompt	70.12	65.85	65.08	55.82

Table 9: Downstream performance under different template variants (25% data, RMI 50-75%; HE = HumanEval).

F On Model-Specific Bias and Self-Reinforcement

RMI scores are inherently model-dependent: different models produce somewhat different rankings, with cross-model Spearman correlations ranging from 0.63 to 0.80. We acknowledge this model-specific bias exists.

However, this is distinct from *self-reinforcement*,

which would require a feedback loop where training amplifies the selection bias. This does not occur for three reasons. First, training optimizes $PPL(A|Q)$, not RMI, which is computed via $P(Q|A)$ —the two objectives are in opposite directions. Second, RMI is a static, one-time metric computed before training begins and is never updated. Third, empirically, RMI rankings remain stable after fine-tuning (Spearman $\rho > 0.9$, see Section 4.3), confirming no systematic shift.

The model-specific bias is therefore static (originating from pretraining) and does not amplify during fine-tuning. Involving the target model’s perspective in data selection is a reasonable design choice, analogous to curriculum learning, where training materials are tailored to the learner’s current knowledge state.

G Computational Cost

Table 10 reports wall-clock time for computing selection scores on the WarriorCoder dataset (310K samples) using $8 \times A100$ GPUs.

Method	Time	Notes
SCAR	~1.5h	Small embedding model
RDS+	~2h	Single pass on Q and A
RMI (single)	~3h	PPL over Q only
IFD	~6h	Two passes over A
QAQ (Ours)	~6h	Two RMI computations

Table 10: Wall-clock time for score computation on WarriorCoder 310K ($8 \times A100$ GPUs).

RMI single-pass is faster than IFD because it operates over questions (Q), which are typically shorter than answers (A) in code datasets. The full QAQ pipeline (Cognitive Gap) requires two model inferences, bringing total selection time to ~6h.